A Reliable Real-Time Slow DoS Detection Framework for Resource-Constrained IoT Networks

Conference or Workshop Item

A Reliable Real-Time Slow DoS Detection Framework for Resource-Constrained IoT Networks

Andy Reed
Sch of Computing & Communications
The Open University
Milton Keynes, United Kingdom
andy.reed@open.ac.uk

Laurence S. Dooley
Sch of Computing & Communications
The Open University
Milton Keynes, United Kingdom
laurence.dooley@open.ac.uk

Soraya Kouadri Mostefaoui
Sch of Computing & Communications
The Open University
Milton Keynes, United Kingdom
soraya.kouadri@open.ac.uk

***Abstract*—Slow DoS attacks have proven to pose a significant security threat to low-resource IoT devices and networks, because they can be launched by nodes which consume nominal bandwidth and have limited resource capability. This makes such malicious attacks easy to initiate, but difficult to mitigate. There also exists the recurrent likelihood of misclassifying legitimate nodes, which are incurring slow or poor network connectivity, as malicious activity. Existing intrusion detection systems (IDS) for detecting Slow DoS attacks often require the creation of large datasets for post event analysis. A functional disadvantage of this dataset-driven approach is the sheer volume of data required, due to the high number of network attributes and events collated, which precludes an in-line, real-time IDS detection solution for live IoT networks. This paper presents an innovative IDS detection framework for resource constrained IoT networks. Using a set of only four attributes, a two-step analysis of live IoT network events enables Slow DoS attacks, in the form of Slowloris, to be both efficiently and reliably detected in real-time. In addition, this lightweight IDS framework can accurately distinguish between malicious and genuine nodes encountering slow or intermittent network connections.**

***Index Terms*—Internet of things, Intrusion detection, Slow DoS, Slowloris**

I. Introduction

Internet of Things (IoT) networks have witnessed rapid growth in adoption as they increasingly offer application-layer services like HTTP (Hyper Text Transfer Protocol) on top of TCP (Transmission Control Protocol) and IP (Internet Protocol). Their ubiquity is appealing by virtue of their ease of operation, mobility, scalability and the generally low cost of IoT devices. Limited resources, however, concomitantly means such devices offer restricted security features so IoT networks are innately vulnerable to malicious attack [1] as highlighted by the Mirai botnet attack which triggered large-scale web server disruption [2].

One such malicious threat to web services is the Denial of Service (DoS) attack, which is designed to flood the target with large amounts of traffic causing excessive use of limited memory and processing resources. These high-volume DoS attacks originate, in many instances, from compromised IoT devices such as CCTV cameras, and webcams [3], with the prevalence of these attacks being analysed by [4], [5]. This type of security threat, where multiple malicious devices launch coordinated network attacks is generally referred to as Distributed DoS (DDoS).

Various IoT intrusion detection systems (IDS) have been proposed to detect DoS and DDoS attacks [6], with most identifying high incoming traffic volume as characteristic of malicious activity. This has led to several, open-source IDS datasets [7] being created, however these contain huge amounts of post-event network data, with the corollary being that the resulting high storage and processing requirements restrict their usage to only offline analysis. This presents significant challenges to utilising them for real-time IDS in IoT networks, given the limited resources capability of the constituent devices [8].

To evade conventional IDS mechanisms, which are tailored to detect abnormally high volumes of network traffic, attackers utilise the Slow DoS attack which is a more subtle DoS variant. This is a legitimate request for application layer services, but with hostile intent. This makes Slow DoS traffic patterns hard to discriminate from legitimate traffic [9] so making reliable and accurate detection difficult. Conventional IDS scan for anomalies like malformed packets or high traffic bursts received over a short time frame, but such approaches are ineffectual for Slow DoS detection because of the latent danger of misclassifying legitimate traffic as malicious [10] and thus erroneously blocking network access.

Another critical factor compounding the challenge of Slow DoS detection in IoT networks, is the generally low traffic volume and bandwidth requirements, with the corollary being these threats are often wrongly mistaken for genuine nodes with either a poor or slow connection [11], [12]. Slow DoS attacks can be launched from a single threat actor with low bandwidth and processing power, so their potential impact on resource scarce IoT nodes, like border routers running web services, is more severe.

To reliably detect Slow DoS attacks on IoT web servers, this paper presents an innovative real-time IDS strategy framed by a critical analysis of TCP/IP traffic within a bounded live IoT network environment comprising only low-resource nodes. The solution is more computationally efficient than existing dataset-driven IDS approaches because of the small number of attributes employed which leads to a commensurately streamlined IoT-centric dataset. Furthermore, unlike existing

solutions which involve post-event analysis, this method affords real-time detection. The test IoT network environment used comprises three node classes; legitimate nodes (LN), genuine nodes with slow-to-intermittent connections (SN) and malicious nodes (MN) which are responsible for instigating the Slow DoS attacks. Experimental results corroborate that Slow DoS attack traffic can be accurately labelled as suspicious for further scrutiny as part of a two-step IDS strategy which can successfully discriminate between MN, LN and SN traffic, thereby improving the reliability of attack detection in resource constrained IoT environments, while incurring minimal overheads.

The remainder of this paper is organised as follows: Section 2 reviews both the nature and characteristics of Slow DoS attacks, along with current detection methods and IDS strategies. Section 3 details the experimental test IoT environment used, with Section 4 presenting a critical results analysis of the new Slow Dos detection technique. Section 5 provides some concluding comments and further work plans.

II. Related work

Slow DoS attacks are amongst the most challenging IoT network security treats because they focus on popular HTTP servers by exploiting inherent vulnerabilities in key operational TCP and HTTP parameters. The operation of TCP is instrumental in realising a Slow DoS attack because it is a connection-orientated protocol, where each successful connection undergoes a three-way handshake. Once this process has been completed, the client and server requests and responses are initiated to set the parameters required to exchange data. The server will wait until the application completes the required tasks, or until a local timeout expires before closing the TCP connection. It is specifically this connection orientation and ordered delivery structure that Slow DoS attacks are designed to exploit. For example, Slow DoS can exploit the HTTP GET request by omitting the character string that signals to the server to close the session. By omitting this string from the client request, the server then must expend unnecessary resources waiting for a client response.

A. Slow DoS threats

Slow Dos attacks have generated considerable interest [11] with several mitigation strategies proposed [10], however a recurring finding is that these types of attacks are difficult to reliably and accurately detect as they can be characterised as legitimate users encountering either poor bandwidth or intermittent node-to-node connectivity, which leads to these genuine nodes being falsely classed as a Slow DoS attack by conventional IDS. The incentive for the Slow DoS attacker is to saturate the target web server with legitimate requests and expend the available server resources, thus disrupting operational service. The Apache HTTP web server is one of the most common Internet servers [13] due to its size and ease of installation, so is an ideal choice for IoT devices requiring a web-based interface. To launch a malicious attack on such web servers, several Slow DoS variants have emerged with the principal ones being:

- **Slow Read**: this attack exploits the TCP window size parameter, so any mismatch in the agreed size means the attacking client will read responses from the server very slowly [14] thereby degrading performance.
- **RUDY or Slow HTTP Post**: this attack sends the POST data as the message body which is sent back to the server at a very slow rate, which can be as small as a single byte per minute [15]. RUDY targets the thread-based functionality of web servers by occupying all the available sockets.
- **Range attacks**: this targets the vulnerability of the range request function of the HTTP server. The range value is specified in bytes, i.e., 0-50 and the attacker compromises the range request header by requesting a long stream of bytes, where some are illegally overlapping so forcing the server to waste resources.
- **Slowloris**: the attacker sends partial HTTP GET requests, and the server opens a connection, but the attacker deliberately fails to respond to the server to complete the connection and holds the socket open until the timeout value is reached [16]. Multiple connection requests can then occupy all the available web server sockets. Given this Slow DoS variant is often mistaken for genuine network activity when LN have either poor or slow connections, the remainder of this paper will focus on this Slow DoS attack, with the next section reviewing current detection approaches for particular this threat.

B. Slow DoS Detection Techniques

Many Slow DoS detection methods which have been developed use machine learning (ML) techniques [17], though these inevitably tend to involve very large datasets with a proportionally high numbers of network event attributes.

One of key drawbacks of these ML approaches is their associated processing overheads which are often well beyond the capacity of typical IoT network nodes. Furthermore, the ML approach is reliant on historical network data for training and evaluation [18].

The performance of ML classifiers including Naïve Bayes, Random Forrest, Decision Tree, K Nearest Neighbour and the Multilayer Perceptron in detecting DDoS attacks has been analysed in [9]. This evaluation was performed off-line on the massive CIDDS-001 (Coburg Intrusion Detection) dataset, which is a compendium of over 32 million network events including benign and malicious traffic. However, only a portion of the dataset (1.5 million network events) were extracted for analysis, so the IDS results cannot be pragmatically applied for evaluation purposes in live real-world IoT settings.

An alternative approach in [11] involved creating Slowloris attack detection alerts from PCAP (Packet Capture) event analysis using the minimum value of incomplete HTTP GET requests. Although this set-up used a live environment, only simulated traffic scenarios were considered. Packet analysis has also been used in a signature-based technique [19] to detect

DoS attacks in live IoT networks, however as 14 different traffic attributes are evaluated, it is a computationally intensive solution best suited to off-line, post event detection.

In contrast, TCP/IP packet analysis has received considerable attention for Slow DoS detection, with [4] proposing an alternative to unreliable signature-based IDS, by measuring the expected packet size of the TCP/IP streams. A limitation of this approach is again the large dataset size used [7] which is incompatible with resource constrained IoT networks. A full packet capture approach for low-rate DoS attack detection has been proposed in [4] by creating a dataset of live traffic for ML analysis, though the resulting dataset only considered malicious Slow DoS and legitimate traffic and used numerous unidentified attributes necessitating a server with 32GB of memory.

Table 1 summarises the range of different Slow DoS attack attributes used in existing open source and purpose-built datasets. Often all these attributes are utilised which inevitably leads to large datasets that incur significantly more memory and processing overheads than is typically available for IoT nodes. Interestingly, while there is notable commonality in many of the attributes used, neither packet length nor TCP delta times are widely employed. Also, most of these dataset-driven solutions involve post event detection rather than in-line, real time identification, which is a more propitious strategy for resource scarce IoT nodes.

TABLE I
COMMON SLOW DOS ATTACK DATASET ATTRIBUTES.

Attribute Name	Staiwan 2019	Baig 2020	liu 2020	Idhammad2017
IP Source	✓	✓	✓	✓
IP Destination	✓	✓	✓	✓
Source Port	✓	✓	✓	✓
Destination Port	✓	✓	✓	✓
IP TTL (Time to Live)	✓			
Packet length		✓		
Frame Length	✓		✓	✓
Frame Number (Count)	✓			✓
IP Protocol	✓		✓	✓
IP Length	✓	✓	✓	
IP Flags	✓			✓
TCP Segment Length		✓	✓	
TCP Header Length	✓	✓		✓
TCP Window Size	✓			✓
TCP Delta Time		✓		

This provided the motivation to investigate an efficient approach to detecting Slowloris attacks in IoT environments, by both minimising the attributes requirement and considering real-time live IoT traffic scenarios, where as well as LN and MN, the causal impact on poor connection SN is critically appraised. The outcome is an elegant, streamlined IoT-centric dataset which includes only those TCP/IP attributes sufficient, yet necessary, to reliably discriminate between benign and malicious inbound HTTP GET requests, thus ensuring the solution is computationally lightweight to operate in resource constrained IoT scenarios. The next section describes the technical details of the experimental live IoT environment which was employed.

III. EXPERIMENTAL IOT NETWORK

The experimental IoT network topology shown in Fig. 1 has been synthesised to generate real time live traffic in a similar way to [20]. It comprises five nodes; two LN, one SN and a MN which generates Slowloris attack traffic targeting port #80 of an Apache web server. The respective technical details of the network devices used for extraction and analysis are provided in Fig. 1. While a tightly constrained test scenario, this topology can be easily scaled beyond the current size with the inclusion of additional subnetworks. The web server runs Ubuntu server 20.04.1 LTS with Apache 2.4 installed, so it has a lightweight operating system appropriate for IoT environments. All tests are carried out on the webserver based on its default security and operational settings. To ensure consistency in all the experiments, each set of tests and observations for SN, LN and MN activity were recorded over a period of 420 s.

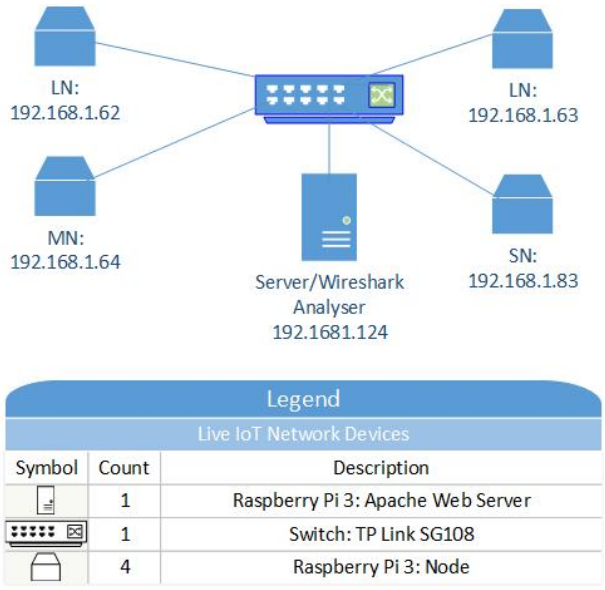

Fig. 1. Live IoT Network Topography.

Malicious traffic is generated from a SlowHTTPtest utility with default installation settings on the MN to create the successful Slowloris attack [20], [21], while for the SN, the throttle feature of the Google developer tool is utilised to enforce an indicative latency with intervals between 1500 ms and 3000 ms, each with 100 kbps upload and download bitrates to simulate IoT devices incurring poor or intermittent 2G connectivity.

For the TCP/IP capture and analysis, Wireshark is configured on the network interface card of RPi 4 (Web server) to extract network packets in PCAP format [21], which is an industry standard application programming interface designed to reliably capture live network data packets [22].

A key feature of the extracted dataset from this IoT test environment is the small number of attributes analysed, which implies lower resources being incurred, in contrast to the large dataset examples in Section II.

For accurate TCP/IP traffic identification, both the IP source address and TCP destination port address attributes are required, which are collectively referred to as classifier attributes as their key function is to label each flow. Since Slowloris is renowned for exploiting the connection session between client and server, the packet length and TCP delta time attributes (see Table 1) are the primary focus of the analysis. Thus, two attributes are used for identification of TCP/IP streams, and two attributes to detect the Slow DoS attack. A comparative analysis of Slow DoS packet lengths is given in [23], whilst [24] evaluated the variance of delta times on the basis that Slowloris exploits the web server wait and timeout parameters. This means the connection must be held open until the timeout value expires, so a longer than normal delta time value is a valuable indicator of a possible Slowloris attack. The next section contextualises the new real-time Slow DoS detection framework, with Slowloris as the MN attack mechanism.

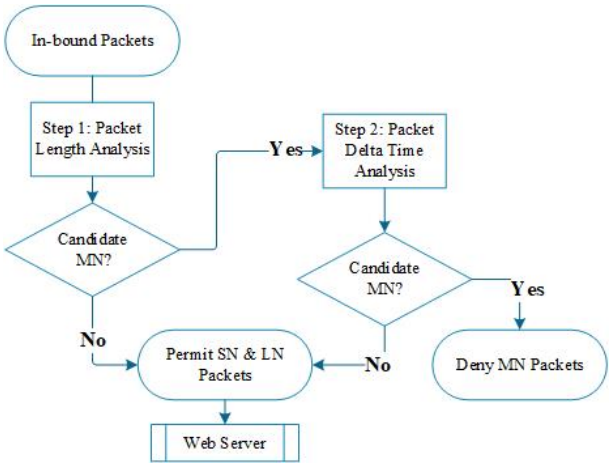

Fig. 2. Slowloris Detection Flowchart

A. Slow DoS Detection Framework

This two-step framework involves just four attributes being extracted at the ingress point of the network to permit real-time packet inspection. This not only expedites MN detection, but avoids the inherent requirement of existing collaborative detection approaches for using post event (off-node) ML-based methods. A flowchart of the detection framework is shown in Fig. 2. Step 1 examines all inbound SN, LN and MN packets destined for port #80, using three attributes. As previously mentioned, two of these relate to packet classification, namely the source IP address and destination port pairs. The third attribute is the packet length *lp* where a range of non-characteristic *lp* values are analysed, and if germane, labelled as *candidate MN* for further scrutiny.

As Slowloris attacks hold TCP connections open without requesting data, MN packets have a smaller payload compared to SN and LN packets, so these have a commensurate impact on the overall packet length. The first detection step thus seeks to identify abnormally sized packets which are labelled as *candidate MN* prior to them undergoing further validation in Step 2 of Fig. 2. The second step applies a fourth attribute to the *candidate MN* list, namely the TCP delta time Δt, which measures the time elapsed between non-contiguous TCP connections. This attribute records the inter-frame arrival times Δt, which is the time frame between the end of one packet and start of the next. Since Slowloris attacks invoke reoccurring connections during an attack period, Δt analysis is used to identify anomalous frequencies in transmissions from the *candidate MN* which have been distilled from Step 1.

IV. Results Discussion

Legitimate TCP/IP *lp* values can be highly variable in nature depending on their origin, the application payload and the underlying protocol used, i.e., HTTP or FTP (File Transfer Protocol). For the analysis here the TCP/IP length, *lp* includes a full byte count of each TCP/IP packet. Fundamental to packet analysis are the SYN (synchronise) and ACK (acknowledge) flags, which are required for creating legitimate TCP bidirectional connections, along with the PSH (push) flag, which informs the server to send data. In the case of a Slowloris attack, once a connection is opened, the MN requests further data with the express purpose of keeping the TCP connection open. Since Slowloris attacks closely resemble LN requests, the first set of experiments investigated the variance in TCP/IP packet lengths that target application port #80 of the web server, so an accurate and reliable detection strategy can be formulated which not only discriminates LN from MN, but crucially, also can differentiate SN from MN.

A. Packet Length Analysis

The corresponding *lp* variance results are displayed in Fig. 3 from which it is observed that packets in the range [40, 79] bytes occur with the highest frequency. This particular range, which contains both SYN and ACK packets, has the highest proportion of traffic, with LN, SN and MN traffic appearing similar, so corroborating the judgement [16] that Slow DoS attack detection for reliable IDS is problematic when attempting to differentiate between LN and MN, with SN often being misclassified as Slow DoS attacks.

A study of the mid-range *lp* values reveals that MN generated packets recorded in the [80, 159] byte range constitute $\approx 13\%$ of all MN traffic, with an average packet length being 96.4 bytes, while for the LN and SN tested, there is a notable absence of packets within this range. By considering the next range of interest, [160, 319] bytes, the corresponding results reveal that $\approx 14\%$ of MN packets lie in this range, in contrast to LN and SN traffic which both have much lower occurrences, with neither exceeding 4%.

When combining these two packet ranges i.e. [80, 319] bytes, MN analysis indicates that $\approx 27\%$ of all packets are recorded in this range, which is a significantly higher than for each LN and SN tested. This provides a heuristic threshold for labelling such traffic as *candidate MN*.

A critical evaluation of the Slowloris attack traffic revealed the presence of open TCP connections where response requests

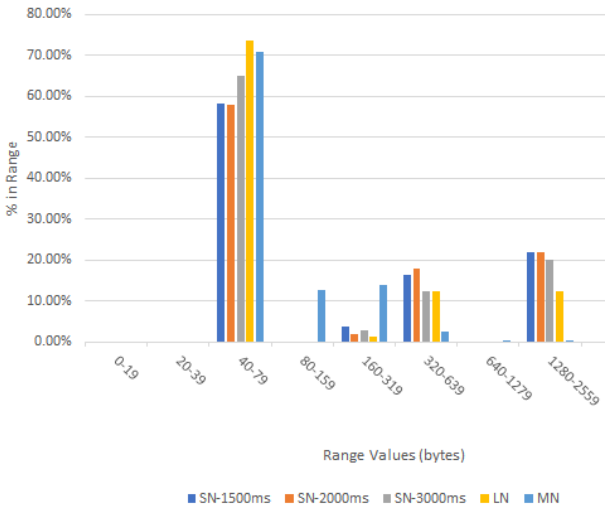

Fig. 3. TCP/IP *lp* Variance Analysis.

from the MN existed within the $80 \leq lp \leq 319$ bytes range. Furthermore, there were excessive TCP retransmissions with flags set to PSH and ACK, with each event recording a packet length $lp = 219$ bytes. As Fig. 4 shows, this is significant in being able to identify potential malicious Slowloris attacks, because it indicates parallel connections have been created to the web server, implying that more data is to follow, and thereby keeping the session open.

```
192.168.1.64        219 35064 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256
192.168.1.64        219 35066 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256
192.168.1.64        219 35068 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256
192.168.1.64        219 35070 → 80 [PSH, ACK] Seq=1 Ack=1 Win=64256
```

Fig. 4. TCP/IP Retransmitted Packets

Classifying abnormally high occurrences of packet lengths within the threshold $80 \leq lp \leq 319$ bytes range affords an effective method to identify *candidate MN*, though there is the possibility some genuine SN will be misclassified if the decision is predicated on this single threshold window. This was the rationale for introducing Step 2 (Fig. 2), where *candidate MN* packets are further inspected to identify suspicious transmission frequencies, with this frequency analysis using a wider observation frame than the *lp* analysis in Step 1.

B. TCP Inter arrival (delta time) analysis

This takes the *candidate MN* distilled in Step 1 and analyses their extracted TCP Δt values. This metric identifies the inter-arrival period of packets from the test dataset and indicates MN activity by a recognisable pattern throughout the attack period. The respective MN, LN and SN Δt plots in Figs. 5, 6 and 7 represent TCP events in 100 ms increments for an assumed connection request interval of 10 s. While the overall volume of traffic during the Slowloris attack is not suspiciously high, there is a discernible increase in the frequency of TCP SYN packets, with these appearing as high peak events. Thus, if P_k is the sequence of high peak events in seconds, $P_k = \{pk_1, pk_2, \ldots, pk_n\}$, then by determining the time intervals between consecutive high peak events, $pk_1, pk_2, \ldots, pk_n$, these can be symptomatic of a recurrent and ordered set of transmissions.

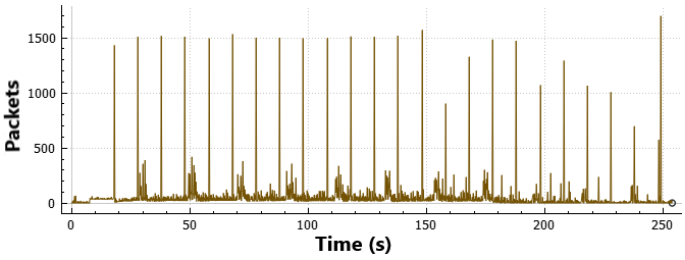

Fig. 5. MN Δt plot

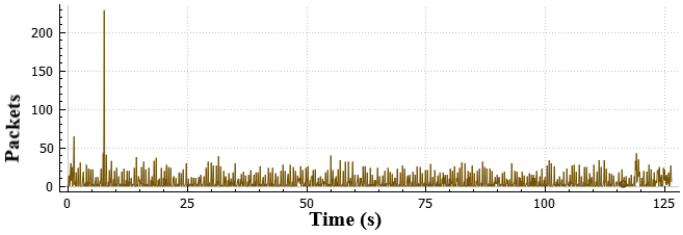

Fig. 6. LN Δt plot

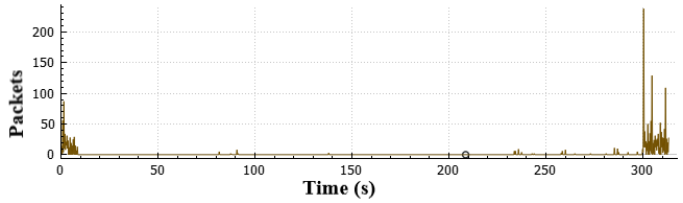

Fig. 7. SN Δt plot

By observing the Δt values in the respective MN, LN and SN plots in Figs. 5, 6 and 7, TCP/IP packets are compared over a window length of 120 s, from which it is readily apparent that a higher Δt mean value is indicative of MN activity. Furthermore, these numerical results are corroborated by the visual method used for contextualising suspicious network behaviour in [25], where anomalous network activity is identified for further inspection.

As an illustrative example, consider the first 10 observations of the *candidate MN* values in the test scenario, with each high peak event having corresponding values $P_K = \{17, 27, 37, 47, 57, 67, 77, 87, 97, 107\}$. This gives a mean value of 62 s for high peak observations, which is used to authenticate the *candidate MN* classifications from Step 1. This contrasts with the corresponding LN and SN mean values, which are both negligibly small ≈ 0.7 s.

The significance of this Δt step is that it not only lowers the misclassification rates for LN, but also importantly SN packets. The high frequency Slowloris TCP SYN packets in Fig. 5. reflect a nominal connection request interval of 10 s, though in practice, the interval period of Slowloris events will be dynamic, so presuming MN Δt values for a given interval

period can constrain the ability to tune the IDS to a predefined event frequency.

For this reason, the values extrapolated in Step 2 are pragmatically applied as the threshold to both validate in real-time, the *candidate MN* list and minimise SN misclassifications. This is a novel feature of this IDS framework, with the *lp* and Δt attributes analysed in a computationally lightweight manner, so in comparison with existing, large-scale datasets and post event ML-driven solutions, this Slowloris detection mechanism is propitious for low-resource IoT networks.

V. Conclusions

This paper has presented an innovative, real-time IDS framework for accurately detecting Slow DoS attacks in resource scarce IoT environments. A two-step analysis of live IoT network events reliably identifies malicious Slow DoS attacks in the form of Slowloris. TCP packets are inspected at the ingress point of the network and *candidate MN* packets labelled for further scrutiny. By capturing and analysing only a small set of key network attributes for classification, namely packet lengths and packet delta times, experimental results verify that Slowloris attacks can be accurately discriminated in real-time from legitimate HTTP requests without the requirement for either massive datasets or post event processing. Crucially, the IDS framework consistently distinguishes malicious from genuine nodes encountering either slow or poor connectivity. To further generalise this detection framework, future work will both scale the live IoT network beyond the current single subnetwork topology along with critically evaluating the efficacy of this lightweight framework for multiple Slow DoS type attacks. It will also develop an adaptive mechanism for tuning the IDS in accordance with prevailing network traffic.

References

[1] M. M. Shurman, R. M. Khrais, and A. A. Yateem, "IoT Denial-of-Service Attack Detection and Prevention Using Hybrid IDS," in *2019 International Arab Conference on Information Technology (ACIT)*. IEEE, Dec 2019, pp. 252–254. [Online]. Available: https://ieeexplore.ieee.org/document/8991097/

[2] G. Graff, "The Mirai Botnet Was Part of a College Student Minecraft Scheme — WIRED," 2017. [Online]. Available: https://www.wired.com/story/mirai-botnet-minecraft-scam-brought-down-the-internet/

[3] A. Procopiou, N. Komninos, and C. Douligeris, "ForChaos: Real time application DDoS detection using forecasting and chaos theory in smart home IoT network," *Wireless Communications and Mobile Computing*, vol. 2019, 2019.

[4] L. Zhou, M. Liao, C. Yuan, and H. Zhang, "Low-Rate DDoS Attack Detection Using Expectation of Packet Size," *Security and Communication Networks*, vol. 2017, 2017.

[5] M. M. Salim, S. Rathore, and J. H. Park, "Distributed denial of service attacks and its defenses in IoT: a survey," *The Journal of Supercomputing 2019 76:7*, vol. 76, no. 7, pp. 5320–5363, Jul 2019.

[6] M. Al Qurashi, C. M. Angelopoulos, and V. Katos, "Efficient Intrusion Detection in Ad-Hoc Networks," in *Science Open*. BCS Learning & Development, Sep 2019. [Online]. Available: https://www.scienceopen.com/hosted-document?doi=10.14236/ewic/icscsr19.15

[7] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani, "Towards a Reliable Intrusion Detection Benchmark Dataset," *Software Networking*, vol. 2017, no. 1, pp. 177–200, Jan 2017.

[8] J. Arshad, M. A. Azad, M. M. Abdellatif, M. H. Ur Rehman, and K. Salah, "COLIDE: A collaborative intrusion detection framework for Internet of Things," *IET Networks*, vol. 8, no. 1, pp. 3–14, 2019.

[9] M. Idhammad, K. Afdel, and M. Belouch, "Detection System of HTTP DDoS Attacks in a Cloud Environment Based on Information Theoretic Entropy and Random Forest," *Security and Communication Networks*, vol. 2018, 2018.

[10] N. Muraleedharan and B. Janet, "Behaviour analysis of HTTP based slow denial of service attack," in *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, vol. 2018-Jan. Institute of Electrical and Electronics Engineers Inc., Feb 2018, pp. 1851–1856.

[11] M. Sikora, T. Gerlich, and L. Malina, "On Detection and Mitigation of Slow Rate Denial of Service Attacks," in *International Congress on Ultra Modern Telecommunications and Control Systems and Workshops*, vol. 2019-Oct. IEEE Computer Society, Oct 2019.

[12] C. L. Calvert and T. M. Khoshgoftaar, "Impact of class distribution on the detection of slow HTTP DoS attacks using Big Data," *Journal of Big Data*, vol. 6, no. 1, pp. 1–18, Dec 2019.

[13] N. R. Fitri, A. H. Budi, I. Kustiawan, and S. E. Suwono, "Low interaction honeypot as the defense mechanism against Slowloris attack on the web server," *IOP Conference Series: Materials Science and Engineering*, vol. 850, no. 1, 2020.

[14] C. Kemp, C. Calvert, and T. M. Khoshgoftaar, "Utilizing netflow data to detect slow read attacks," in *Proceedings - 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018*. Institute of Electrical and Electronics Engineers Inc., Aug 2018, pp. 108–116.

[15] S. Q. Memon, "Prevention Mechanism For RUDY Attack And Its Comparison," *University of Sindh Journal of Information and Communication Technology*, vol. 4, no. 1, pp. 45–51, Mar 2020.

[16] V. d. S. Faria, J. A. Gonçalves, C. A. M. da Silva, G. d. B. Vieira, and D. M. Mascarenhas, "SDToW: A Slowloris Detecting Tool for WMNs," *Information*, vol. 11, no. 12, p. 544, Nov 2020. [Online]. Available: https://www.mdpi.com/2078-2489/11/12/544

[17] J. Liu, B. Kantarci, and C. Adams, "Machine learning-driven intrusion detection for Contiki-NG-based IoT networks exposed to NSL-KDD dataset," in *WiseML 2020 - Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, 2020, pp. 25–30. [Online]. Available: https://doi.org/10.1145/3395352.3402621

[18] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, Nov 2019.

[19] D. Stiawan, D. Wahyudi, A. Heryanto, S. Samsuryadi, M. Y. Idris, F. Muchtar, M. Abdullah Alzahrani, and R. Budiarto, "TCP FIN Flood Attack Pattern Recognition on Internet of Things with Rule Based Signature Analysis," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 15, no. 07, p. 124, Apr 2019. [Online]. Available: https://online-journals.org/index.php/i-joe/article/view/9848

[20] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "FlowGuard: An Intelligent Edge Defense Mechanism against IoT DDoS Attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552–9562, Oct 2020.

[21] Z. A. Baig, S. Sanguanpong, S. Naeem Firdous, V. Nhan Vo, T. Gia Nguyen, and C. So-In, "Averaged dependence estimators for DoS attack detection in IoT networks," *Future Generation Computer Systems*, vol. 102, pp. 198–209, 2020. [Online]. Available: https://doi.org/10.1016/j.future.2019.08.007

[22] T. Keary, "PCAP: Packet Capture, what it is & what you need to know," 2020. [Online]. Available: https://www.comparitech.com/net-admin/pcap-guide/

[23] N. Muraleedharan and B. Janet, "Behaviour analysis of HTTP based slow denial of service attack," *Proceedings of the 2017 International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2017*, vol. 2018-Jan, pp. 1851–1856, 2018.

[24] E. Cambiaso, M. Aiello, M. Mongelli, and I. Vaccari, "Detection and classification of slow DoS attacks targeting network servers," in *ACM International Conference Proceeding Series*. Association for Computing Machinery, Aug 2020.

[25] J. R. Goodall, E. D. Ragan, C. A. Steed, J. W. Reed, G. D. Richardson, K. M. Huffer, R. A. Bridges, and J. A. Laska, "Situ: Identifying and explaining suspicious behavior in networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 204–214, Jan 2019.